

# BLM2041 Signals and Systems

## Syllabus

### The Instructors:

Doç. Dr. Ali Can Karaca  
[ackaraca@yildiz.edu.tr](mailto:ackaraca@yildiz.edu.tr)

Dr. Öğr. Üyesi Erkan Uslu  
[euslu@yildiz.edu.tr](mailto:euslu@yildiz.edu.tr)

# Course Details

- Course Code : BLM 2041
- Course Name:  
Signals and Systems for Computer Engineers  
(Bilgisayar Mühendisleri için Sinyaller ve Sistemler)

Instructor :

1- Doç. Dr. Ali Can Karaca

2- Dr. Öğr. Üyesi Erkan Uslu

# Assesment

Method	Quantity	(%)
Homework	2	20
Quiz	1	10
Midterm Exam	1	30
Final Exam	1	40
Attendance & participation	-	00

By University Rule:

Your average  $< 40$



FF

# Course Outline

## **1. Introduction.**

Mathematical Representation of Signals. Mathematical Representation of Systems.

## **2. Sinusoids.**

Review of Sine and Cosine Functions. Sinusoidal Signals. Sampling and Plotting Sinusoids. Complex Exponentials and Phasors. Phasor Addition. Time Signals.

## **3. Spectrum Representation.**

The Spectrum of a Sum of Sinusoids. Beat Notes. Periodic Waveforms. Fourier Series Analysis and Synthesis. Time-Frequency Spectrum. Frequency Modulation.

## **4. Sampling and Aliasing.**

Sampling. Spectrum View of Sampling and Reconstruction. Discrete-to-Continuous Conversion. The Sampling Theorem.

## **5. Continuous-Time LTI Systems and the Convolution Integral.**

Establishing a General Input-Output Relationship. Working with the Convolution Integral.

## **6. Discrete-Time LTI Systems and the Convolution Sum.**

Specializing the Input/Output Relationship. Working with the convolution Sum.

## **7. LTI System Differential and Difference Equations in the Time Domain.**

Obtaining the differential/difference equations for the input-output relations of systems. Solution of differential and discrete equations in the time domain.

# Spyder (IDE for PYTHON)

The screenshot displays the Spyder Python IDE interface. The main window is titled "Spyder (Python 3.6)" and features a menu bar (File, Edit, Search, Source, Run, Debug, Consoles, Projects, Tools, View, Help) and a toolbar with icons for file operations, running, and debugging. The editor pane shows a file named `tmm_core.py` with Python code for a transfer matrix method (TMM) calculation. The code includes a function `position_resolved` that calculates the Poynting vector, absorbed energy density, and E-field at a specific location. The code is currently at line 405, which is highlighted in pink. The right-hand side of the IDE contains a help panel for the `coh_tmm` function, showing its definition, type, and a detailed description of its parameters. Below the help panel is an IPython console window showing the execution of several commands, including `import tmm`, `temp = tmm.coh_tmm('s', [1, 1.7, 2.1], [inf, 250, inf], 0, 500)`, and `cmath.phase(temp['r'])`. The status bar at the bottom indicates the current line (405), column (11), and memory usage (47%).

```
385 T = (s_data['T'] + p_data['T']) / 2.
386 return {'R': R, 'T': T}
387
388 def position_resolved(layer, distance, coh_tmm_data):
389     """
390     Starting with output of coh_tmm(), calculate the Poynting vector,
391     absorbed energy density, and E-field at a specific location. The
392     location is defined by (layer, distance), defined the same way as in
393     find_in_structure_with_inf(...).
394
395     Returns a dictionary containing:
396
397     * poyn - the component of Poynting vector normal to the interfaces
398     * absor - the absorbed energy density at that point
399     * Ex and Ey and Ez - the electric field amplitudes, where
400       z is normal to the interfaces and the light rays are in the x,z plane.
401
402     The E-field is in units where the incoming |E|=1; see
403     https://arxiv.org/pdf/1603.02720.pdf for formulas.
404     """
405     if layer > 0:
406         v,w = coh_tmm_data['vw_list'][layer]
407     else:
408         v = 1
409         w = coh_tmm_data['r']
410     kz = coh_tmm_data['kz_list'][layer]
411     th = coh_tmm_data['th_list'][layer]
412     n = coh_tmm_data['n_list'][layer]
413     n_0 = coh_tmm_data['n_list'][0]
414     th_0 = coh_tmm_data['th_0']
415     pol = coh_tmm_data['pol']
416
417     assert ((layer >= 1 and 0 <= distance <= coh_tmm_data['d_list'][layer])
418            or (layer == 0 and distance <= 0))
419
420     # Amplitude of forward-moving wave is Ef, backwards is Eb
421     Ef = v * exp(1j * kz * distance)
422     Fh = w * exp(-1j * kz * distance)
```

**coh\_tmm**

**Definition :** `coh_tmm(pol, n_list, d_list, th_0, lam_vac)`

**Type :** Function of `tmm.tmm_core` module

Main "coherent transfer matrix method" calc. Given parameters of a stack, calculates everything you could ever want to know about how light propagates in it. (If performance is an issue, you can delete some of the calculations without affecting the rest.)

`pol` is light polarization, "s" or "p".

`n_list` is the list of refractive indices, in the order that the light would pass through them. The 0'th element of the list should be the semi-infinite medium from which the light enters, the last element should be the semi-infinite medium to which the light exits (if any exits).

`th_0` is the angle of incidence: 0 for normal,  $\pi/2$  for glancing. Remember, for a dissipative medium, `th_0` should be complex, that `th_0` is real

Variable explorer | File explorer | Help

IPython console

Console 2/A

```
In [10]: import tmm
In [11]: temp = tmm.coh_tmm('s', [1, 1.7, 2.1], [inf, 250, inf], 0, 500)
In [12]: (temp['R'], temp['T'])
Out[12]: (0.062435840423349355, 0.93756415957665051)
In [13]: cmath.phase(temp['r'])
Out[13]: 2.751964575481417
In [14]: cmath.phase(temp['r']) / degree
Out[14]: 157.67595554459646
In [15]:
```

Python console | History log | IPython console

Permissions: RW End-of-lines: CRLF Encoding: UTF-8 Line: 405 Column: 11 Memory: 47 %

# Course Outline

## **8. The Fourier Transform for Continuous-Time Signals and Systems.**

Continuous-Time Aperiodic Signals. Continuous-Time Fourier Transform. Properties of Continuous-Time Fourier Transform.

## **9. The Discrete Time Fourier Transform for Discrete-Time Signals.**

Discrete-Time Aperiodic Signals. Discrete-Time Fourier Transform. Properties of Discrete-Time Fourier Transform

## **10. The Laplace Transform for Continuous Time.**

Laplace Transform. Common Laplace Transforms. Properties Of the Laplace Transform. Inverse Laplace Transform. Poles and Zeros in the s-plane.

## **11. The Z Transform for Discrete Time.**

Z Transform. Common Z Transforms. Properties Of the Z Transform. Inverse Z Transform. Poles and Zeros in the z-plane.

# COURSE OBJECTIVES

- Students will be able to:
- Understand **mathematical** descriptions of Signals and Systems
- Express those descriptions as computer **implementations** (MATLAB, OCTAVE, SCILAB, R, PYTHON )
  - Yıldız Technical University provides MATLAB License.
  - OCTAVE, SCILAB, R and PYTHON are free

# COURSE OBJECTIVES

- MATLAB
  - <https://www.mathworks.com/>
- SCILAB
  - <https://www.scilab.org/>
- OCTAVE
  - <https://www.gnu.org/software/octave/>
- R
  - <https://www.r-project.org/>
- PYTHON
  - <https://www.python.org/>



# SCILAB environmet

The screenshot displays the SCILAB environment interface, which includes a file navigator, a central console window, and a variable navigator.

**Console Scilab:**

```

Initialisation :
Chargement de l'environnement de travail

-->a=rand(10,10)
a =

    column 1 to 5
    0.2113249    0.5608486    0.3076091    0.5015342    0.2806698
    0.7560439    0.8623569    0.9329616    0.4368588    0.1280058
    0.8002211    0.7263507    0.2346008    0.2693125    0.7783129
    0.3303271    0.1985144    0.332642    0.6325745    0.2119030
    0.8653811    0.5442573    0.3816361    0.4051954    0.1121355
    0.6283918    0.2320748    0.2922267    0.9184700    0.6854896
    0.8497452    0.2312237    0.5664249    0.8407334    0.1531217
    0.6957110    0.2184633    0.4826472    0.4818509    0.6978851
    0.8782165    0.8833888    0.3321739    0.2639556    0.8415518
    0.2683740    0.6525135    0.5935095    0.4148104    0.4062025

    column 6 to 10
    0.4094825    0.3873779    0.5378230    0.5878720    0.6488563
    0.8784126    0.9222899    0.1199924    0.4828179    0.9923191
    0.1138360    0.9488184    0.2256303    0.2232865    0.0500420
    0.1998338    0.3435337    0.6274093    0.8400886    0.7485507
    0.5418661    0.3760119    0.7608433    0.1205996    0.4104059
    0.5896177    0.7340941    0.0485566    0.2855344    0.6084524
    0.6853380    0.2615761    0.6723950    0.8607515    0.8544211
    0.8906225    0.4993494    0.2017173    0.8494102    0.0642647
    0.5042213    0.2618378    0.3911574    0.5257081    0.8279081
    0.3493815    0.5253563    0.8380317    0.9931210    0.9262344

-->|
    
```

**Navigateur de variables:**

Nom	Dimension	Type	Visibilité
a	10x10	Double local	
home	1x1	Chaine de... local	
PWD	1x1	Chaine de... local	
%k	1x1	Boolean local	
%f	1x1	Boolean local	
%t	1x1	Boolean local	
%nan	1x1	Double local	
%inf	1x1	Double local	
SCI	1x1	Chaine de... local	
SCHOME	1x1	Chaine de... local	
TMPDIR	1x1	Chaine de... local	
%pi	1x1	Boolean local	
%ffw	1x1	Boolean local	
%t	1x1	Boolean local	
%f	1x1	Boolean local	
%eps	1x1	Double local	
%io	1x2	Double local	
%i	1x1	Double local	
%e	1x1	Double local	
%pi	1x1	Double local	
%modaWa...	1x1	Boolean global	
%poolboxe...	1x1	Double global	
%poolboxe...	1x1	Chaine de... global	

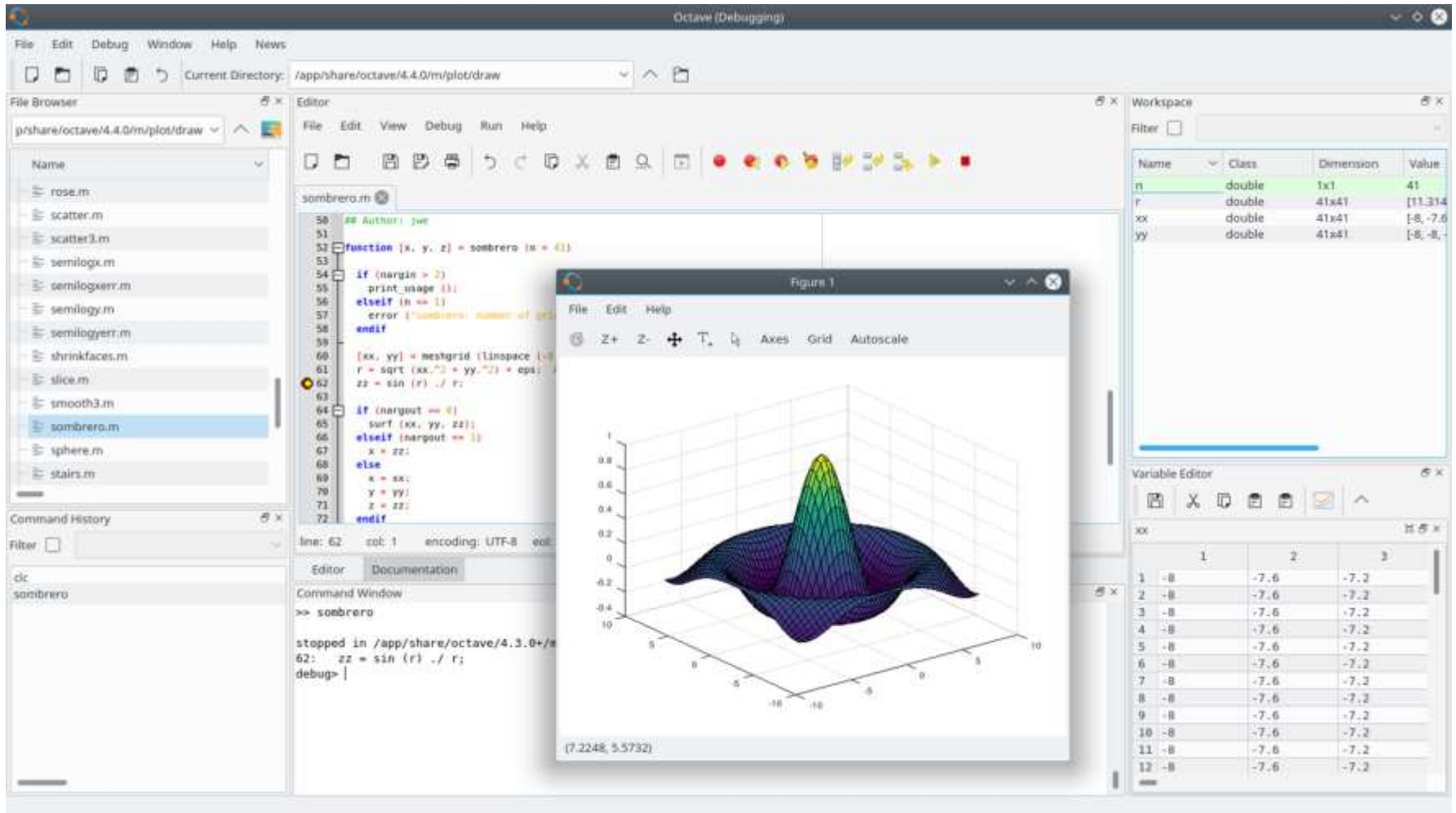
**Historique des commandes:**

```

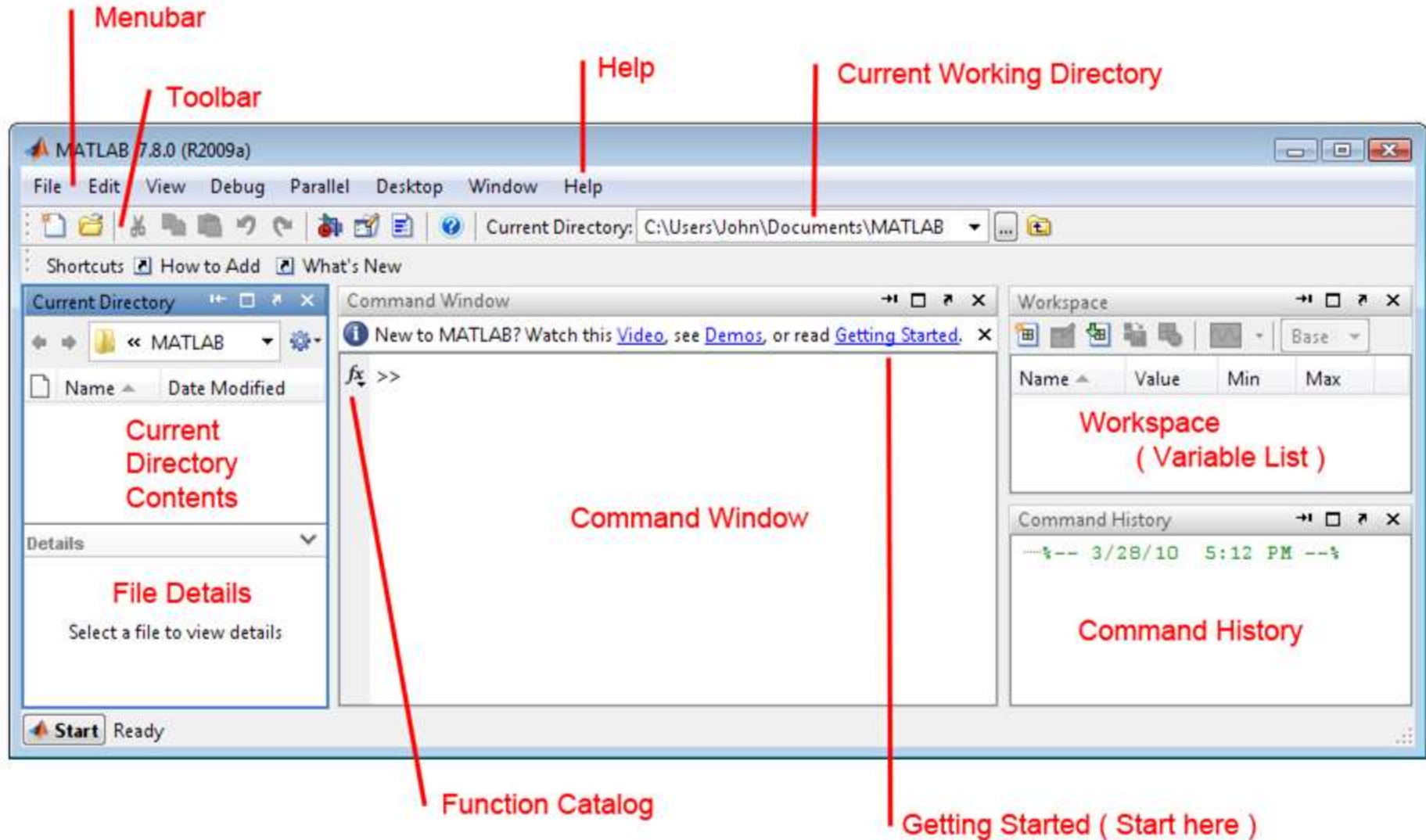
// -- 29/08/2012 09:25:18 -- //
// -- 29/08/2012 11:36:14 -- //
a=rand(10,10)
    
```

At the bottom of the console, there are checkboxes for "Respecter la casse" (checked) and "Expression..." (unchecked).

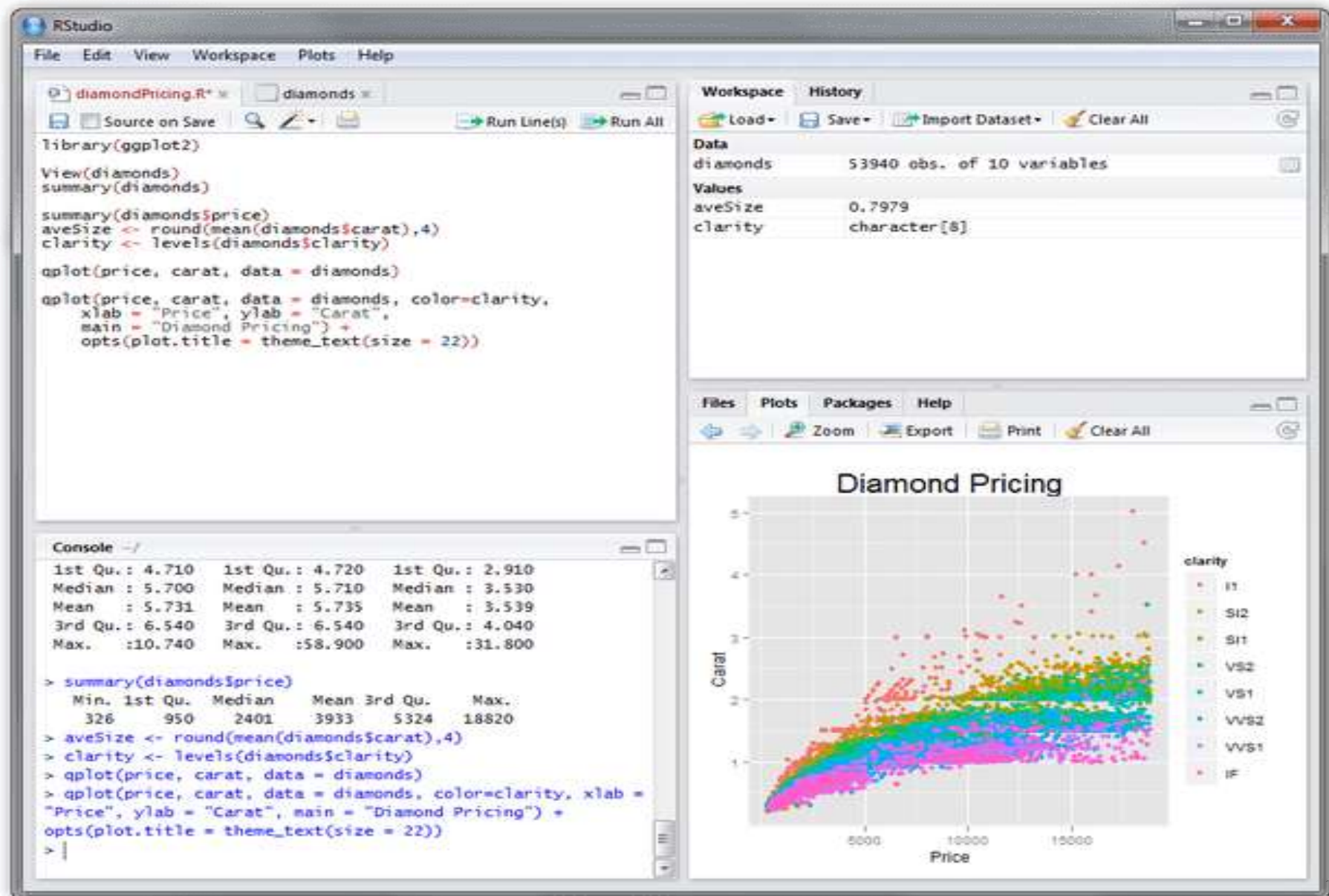
# OCTAVE environment



# MATLAB environment



# Rstudio (IDE for R)





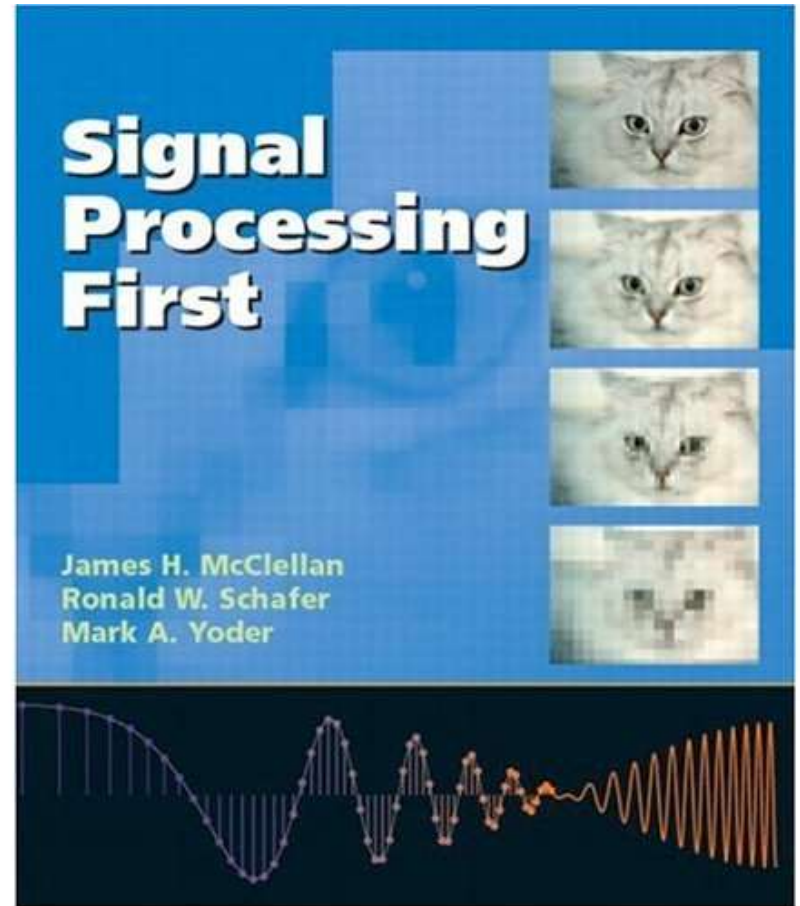
# Main course book

## Signal Processing First

by James H McClellan,  
Ronald W. Schaffer  
and Mark A. Yoder.

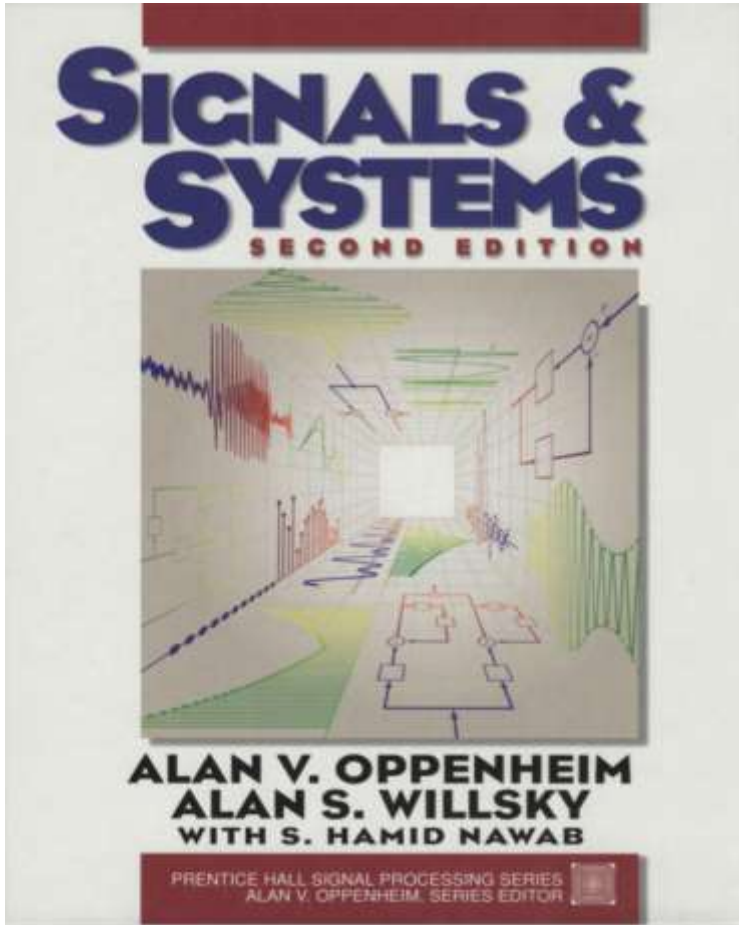
Published by Prentice  
Hall.

Isbn: 0-13-120265-0



# Some Other Books

- by [Alan. V. Oppenheim](#) and [Alan S. Willsky](#)



Oppenheim, Alan V., Alan S. Willsky, and Syed Hamid Nawab. "Signals and systems 2nd ed." *New Jersey: Prentice Hall*(1997).

# Course Objectives (In details)

## Academic knowledge

- Students will be able to:
  - Understand and develop simple mathematical models for representing signals and systems
  - Understand the relationship between time and frequency domain models of dynamic systems
  - Convert time to frequency-domain models and vice versa
  - Understand the relationship between continuous and discrete-time models

## Intellectual skills

- Students will be able to:
  - Build a mathematical model from a real-life problem related to signals and systems
  - Interpret results achieved by mathematical solutions

## Practical skills

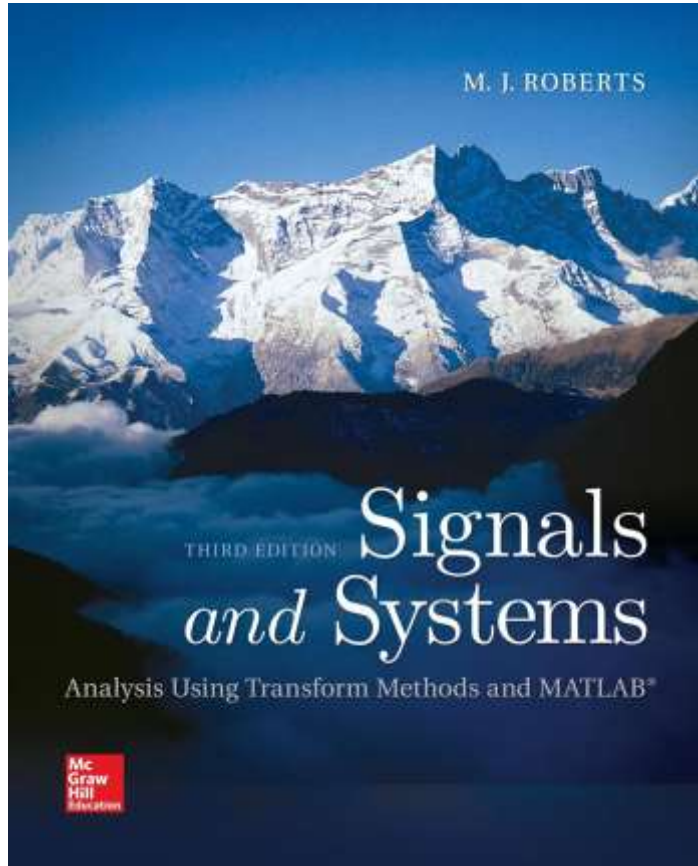
- Students will be able to:
  - Express models and methods as computer implementations (MATLAB or OCTAVE)
    - Yıldız Technical University provides MATLAB License.
  - Apply Matlab/Octave for analysis and simulation of continuous and discrete time systems
  - Analyse mathematical solutions in the context of the original problem

## Transferable skills

- Students will be able to:
  - Choose appropriate approach in problem solving situation
  - Present and communicate formalised results and conclusions

# Some Other Books

- by M.J. Roberts

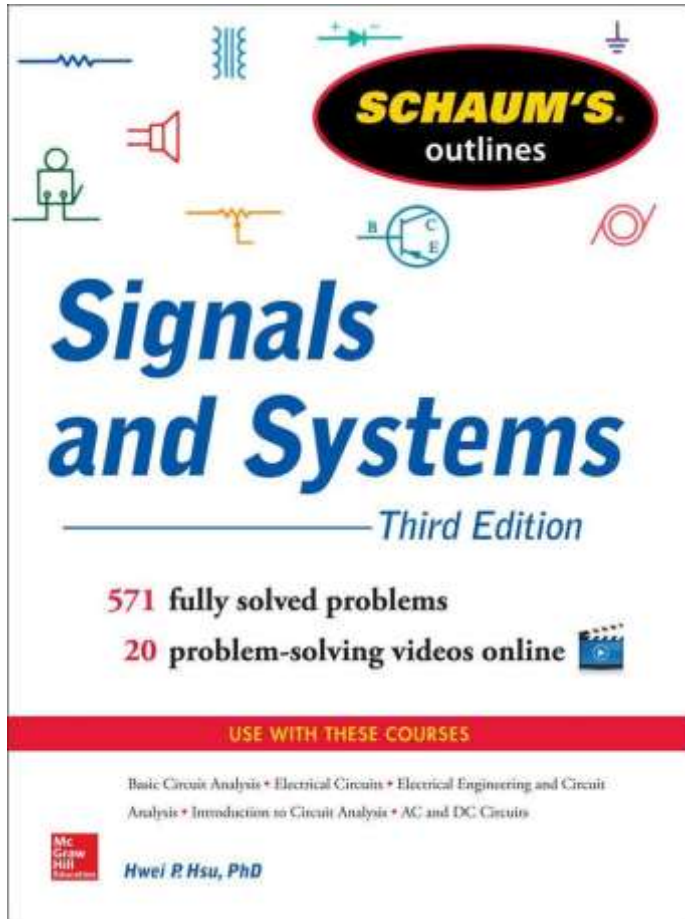


Signals and Systems: Analysis Using Transform Methods & MATLAB



# Some Other Books

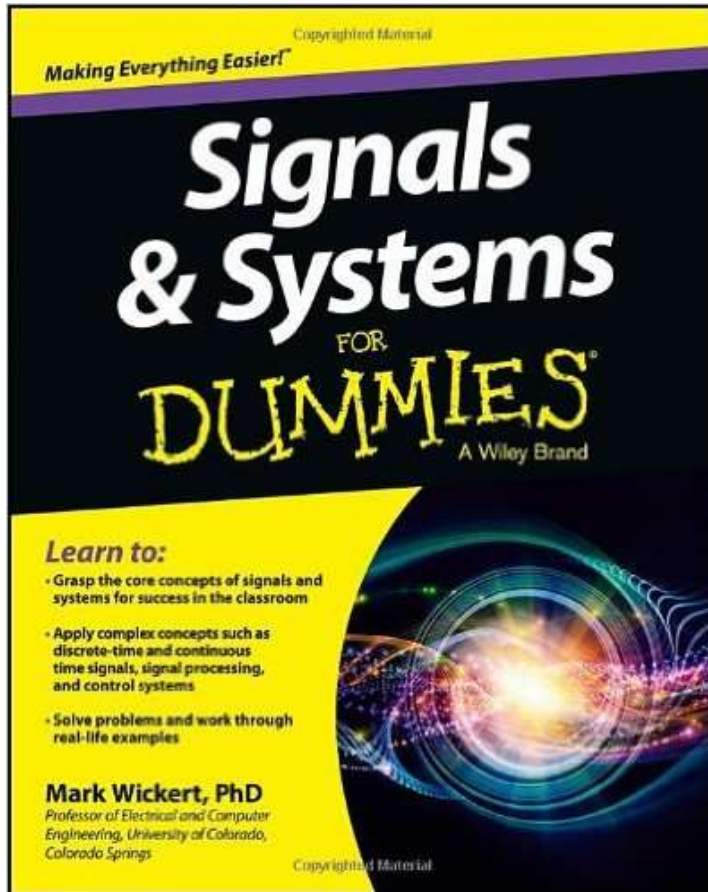
- by Hwei Hsu



Schaum's Outline of Signals and Systems, 2nd or 3rd Edition

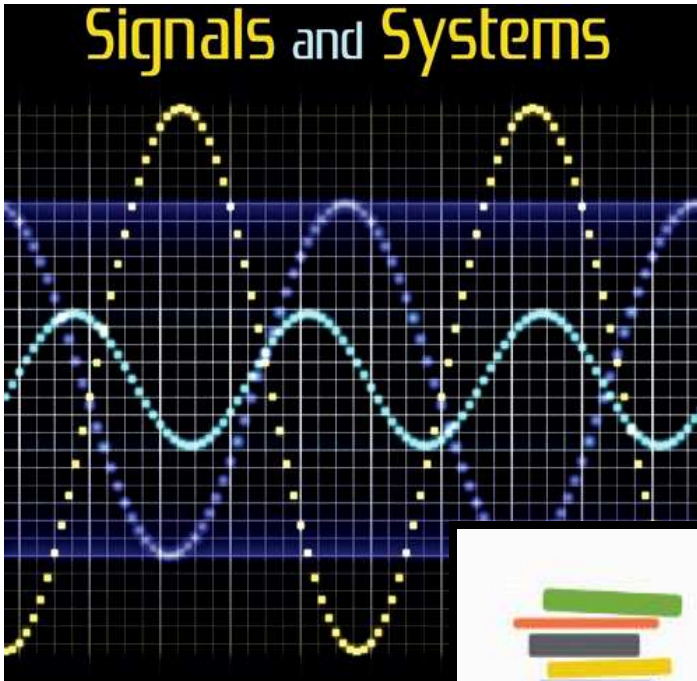
# Some Other Books

- by Mark Wickert



Wickert, Mark. *Signals and Systems for Dummies*. John Wiley & Sons, 2013.

# Some Other Books



- Online e-book by [Richard Baraniuk](#)



<https://cnx.org/contents/d2CEAGW5>

# Rules of the Conduct

- No eating /drinking in class
  - *except water*
- Cell phones must be kept outside of class or switched-off during class
  - *If your cell-phone rings during class or you use it in any way, you will be asked to leave and counted as unexcused absent.*
- No web surfing and/or unrelated use of computers,
  - *when computers are used in class or lab.*

# Rules of the Conduct

- You are responsible for checking the class web page often for announcements.
- Academic dishonesty and cheating will not be tolerated and will be dealt with according to university rules and regulations
  - *Presenting any work, or a portion thereof, that does not belong to you is considered academic dishonesty.*
- University rules and regulations:
  - <http://www.ogi.yildiz.edu.tr/category.php?id=17>
  - [https://www.yok.gov.tr/content/view/544/230/lang,tr\\_TR/](https://www.yok.gov.tr/content/view/544/230/lang,tr_TR/)

# Attendance Policy

- The requirement for attendance is **70%**.
  - Absent in at least two exams ☐ F0